Mass storage interface LTSM for FAIR Phase 0 data acquisition

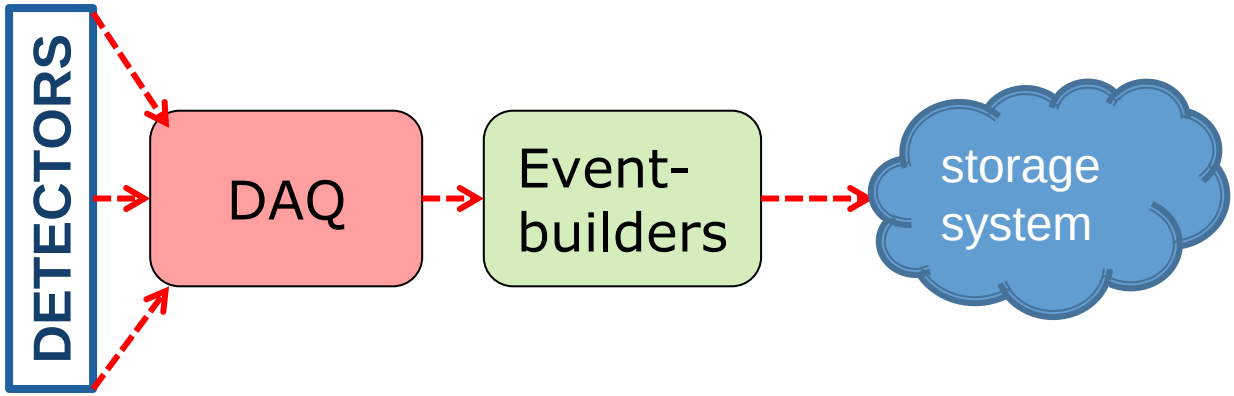
CHEP 2019, Adelaide

Jörn Adamczewski-Musch, GSI / EEL, HADES
Thomas Stibor, GSI / HPC

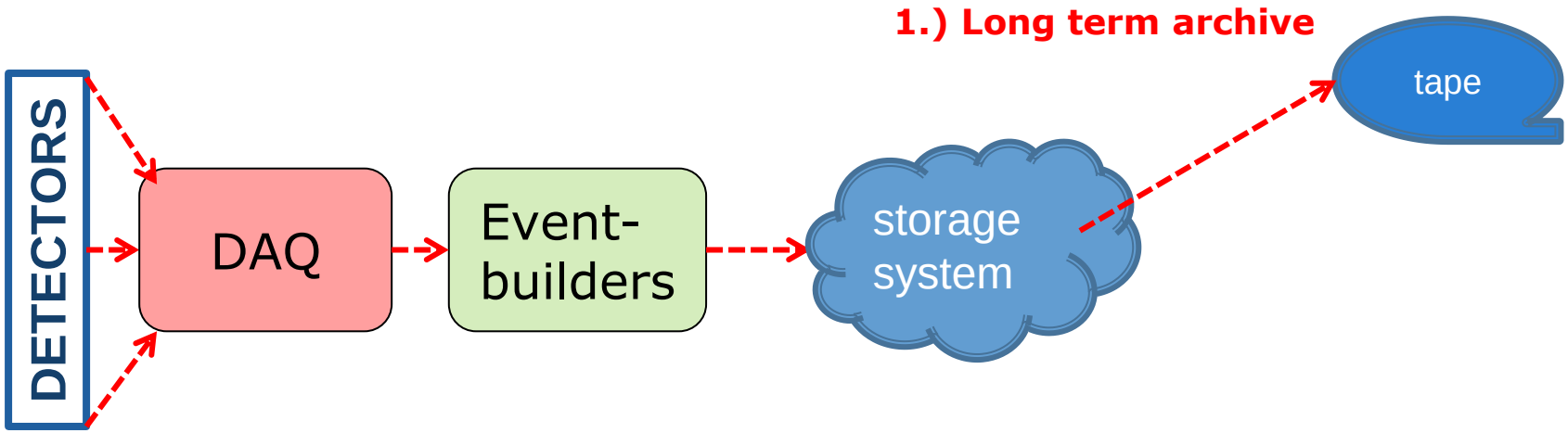
Overview

- Motivation: DAQ storage use cases
- LTSM software
- LTSM with DABC and MBS DAQ
- HADES beamtime 2019 experiences
- Outlook: LTSM with FSD server

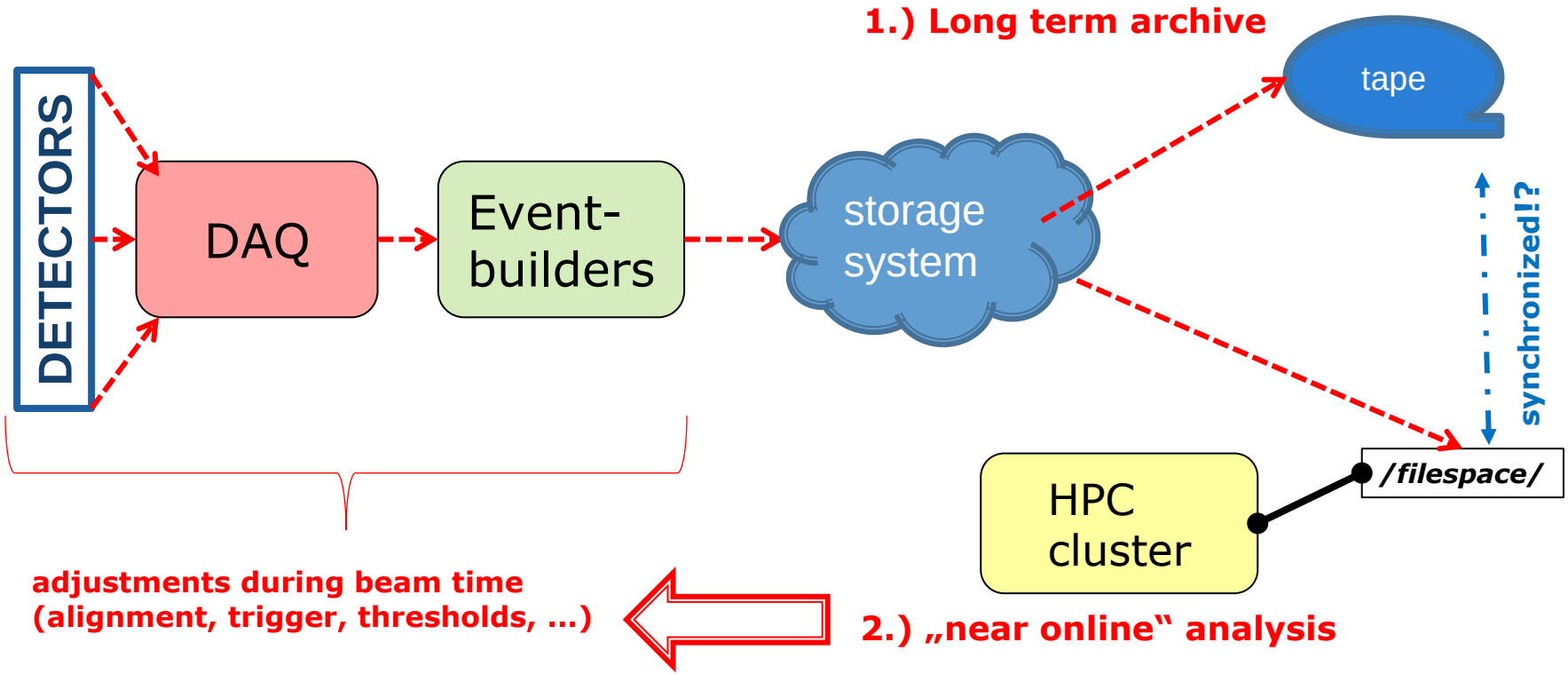
DAQ storage use cases



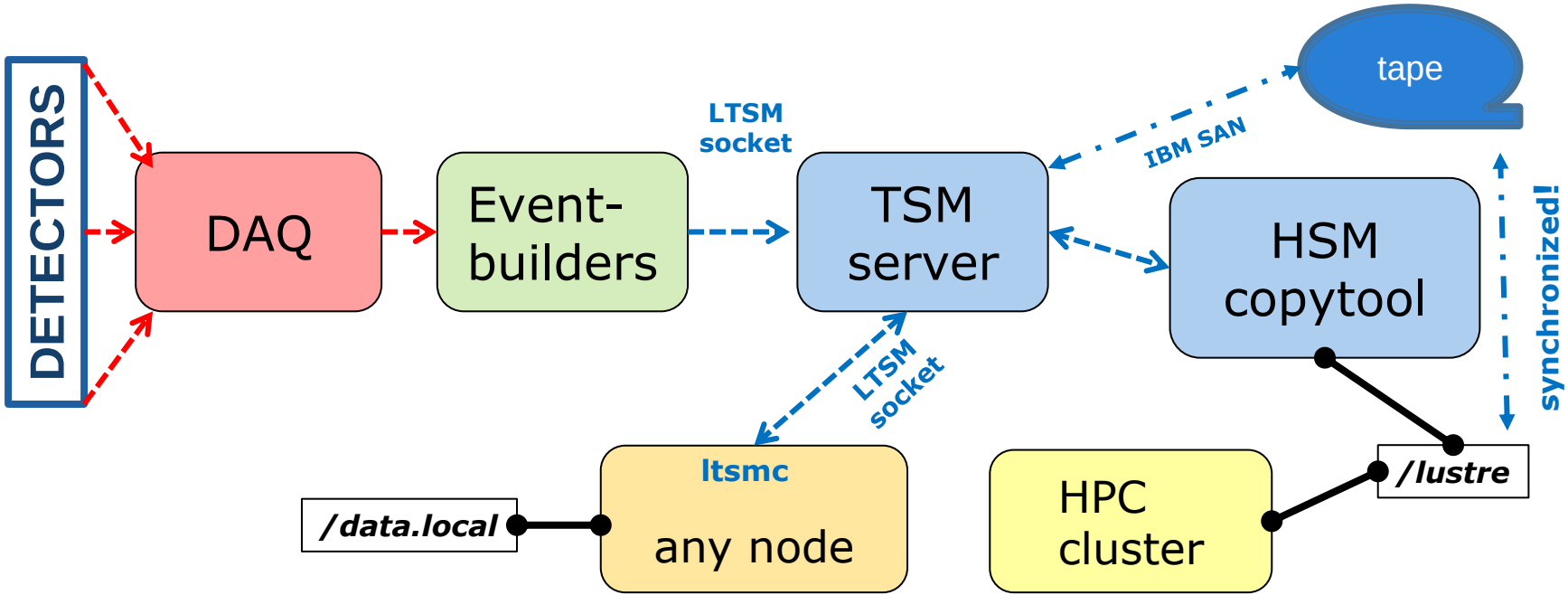
DAQ storage use cases (1)



DAQ storage use cases (2)



DAQ storage with LTSM



The LTSM software (Lightweight Tivoli Storage Management)

Development by Thomas Stibor, GSI-HPC (https:/github.com/tstibor/ltsm)

- Based on IBM TSM client api for x86_64 Linux

1. **Write from client to tape archive via TCP/IP sockets (C-API)**
2. **Transparent synchronization between tape and lustre file system:**

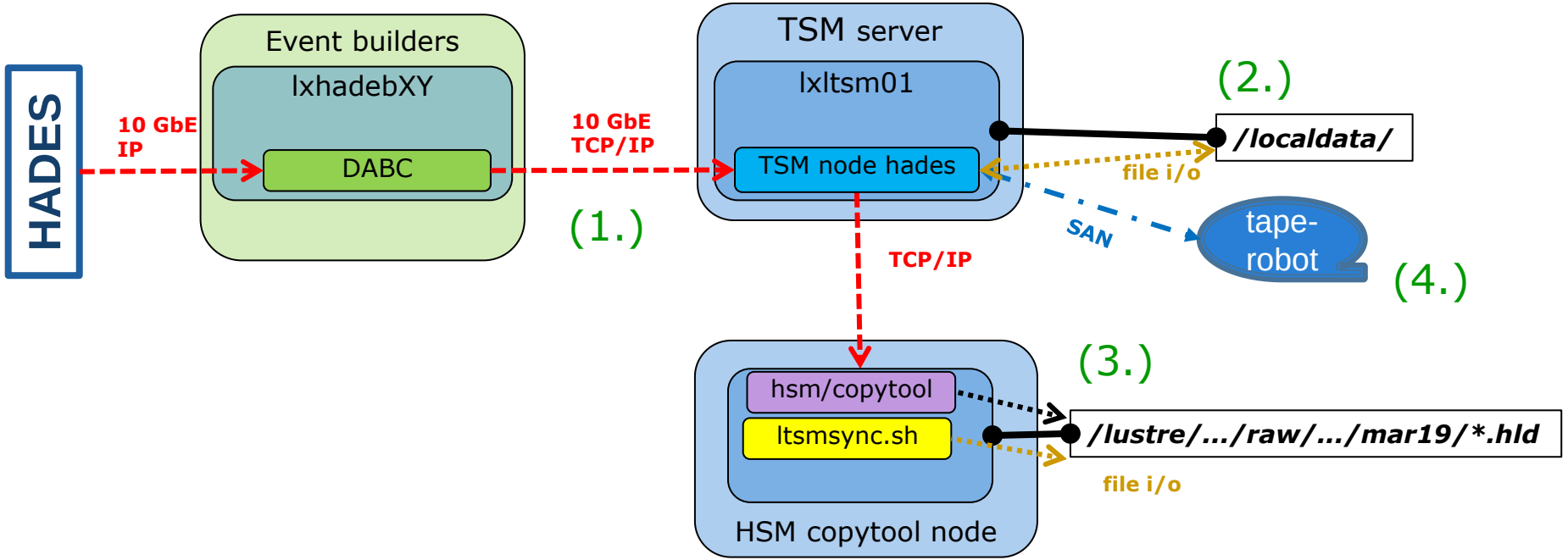
   Lustre HSM (Hierarchical Storage Management) copytool

   - archive file path is same as lustre file path *(*e.g. */lustre/hades/mar19/*.hld*)
   - file can be **retrieved from tape on first user read access** to lustre
   - file can be automatically **archived to tape** from lustre
3. **Command line tool ltsmc**
   - ***query, retrieve, archive, delete*** operations from any TSM client
   - for interactive console or scripting use case

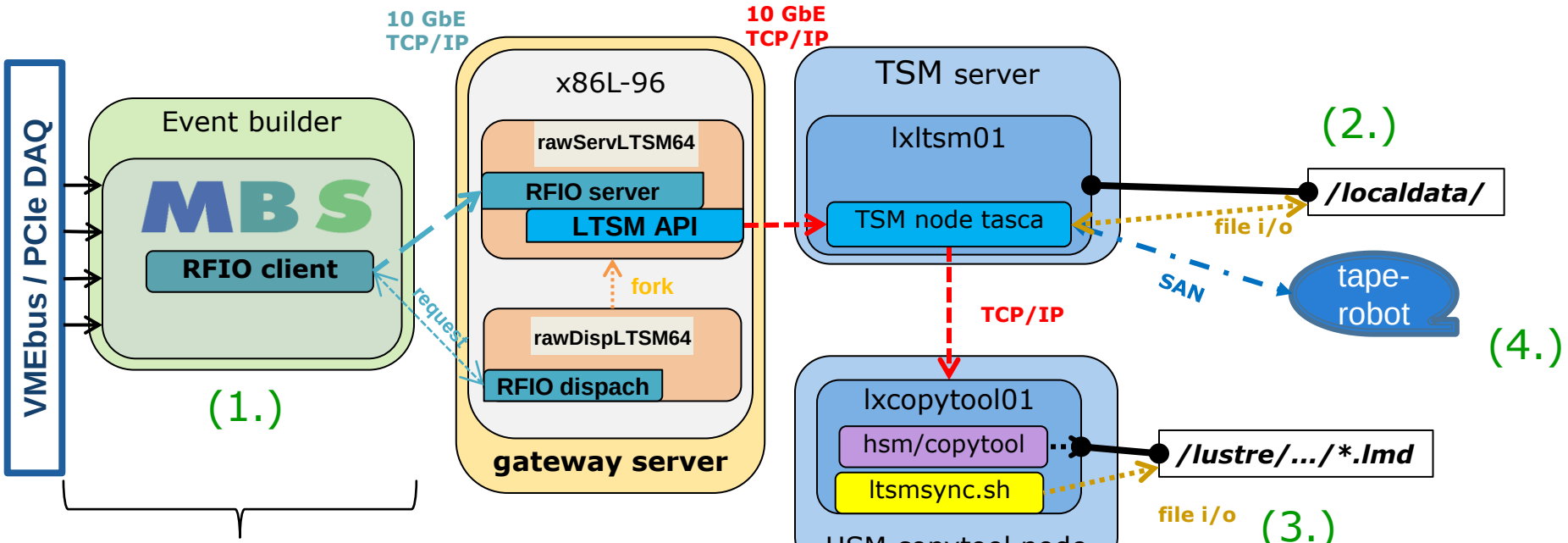
LTSM with FAIR-0 DAQ software

- Data Acquisition Backbone Core (http://dabc.gsi.de)
  - DAQ for **HADES experiment** (http://hades.gsi.de)
  - New: LTSM plug-in with POSIX-like file API for all data formats
- Multi Branch System (http://daq.gsi.de)
  - DAQ for **NUSTAR** and **APPA** research pillars of FAIR
  - New: Interface server between old gstore protocol and LTSM

LTSM for HADES with DABC plug-in

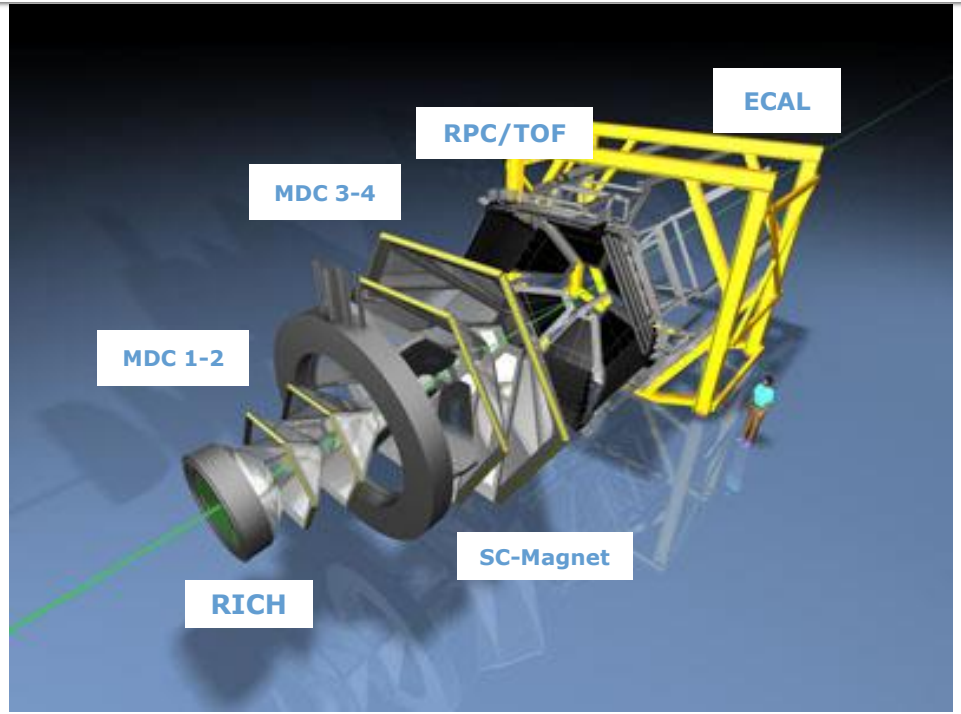


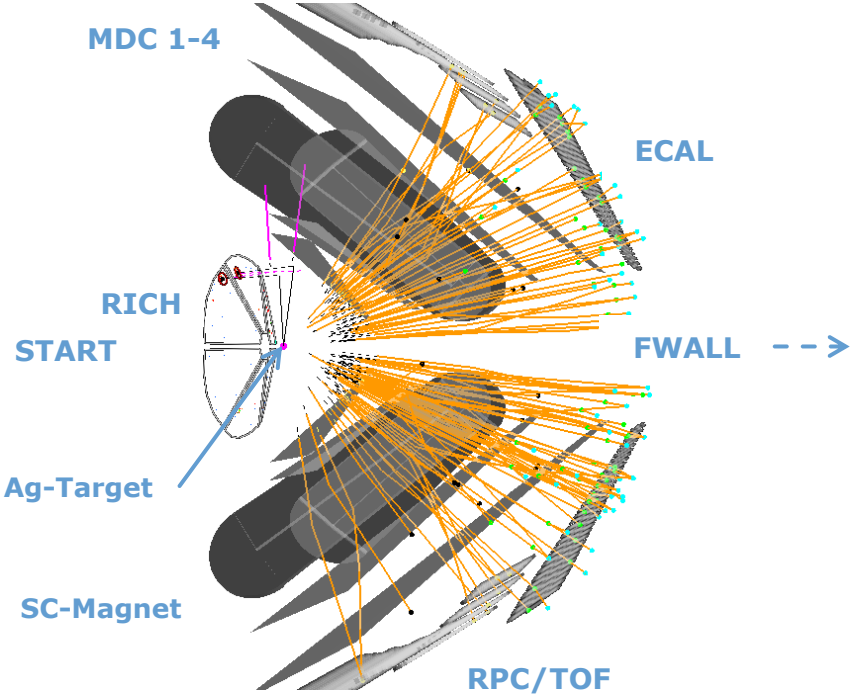
LTSM for MBS DAQ with RFIO gateway



MBS node does not need TSM client library!
(<- not available for LynxOS, PPC VME Linux etc.)
use legacy gstore/RFIO protocol

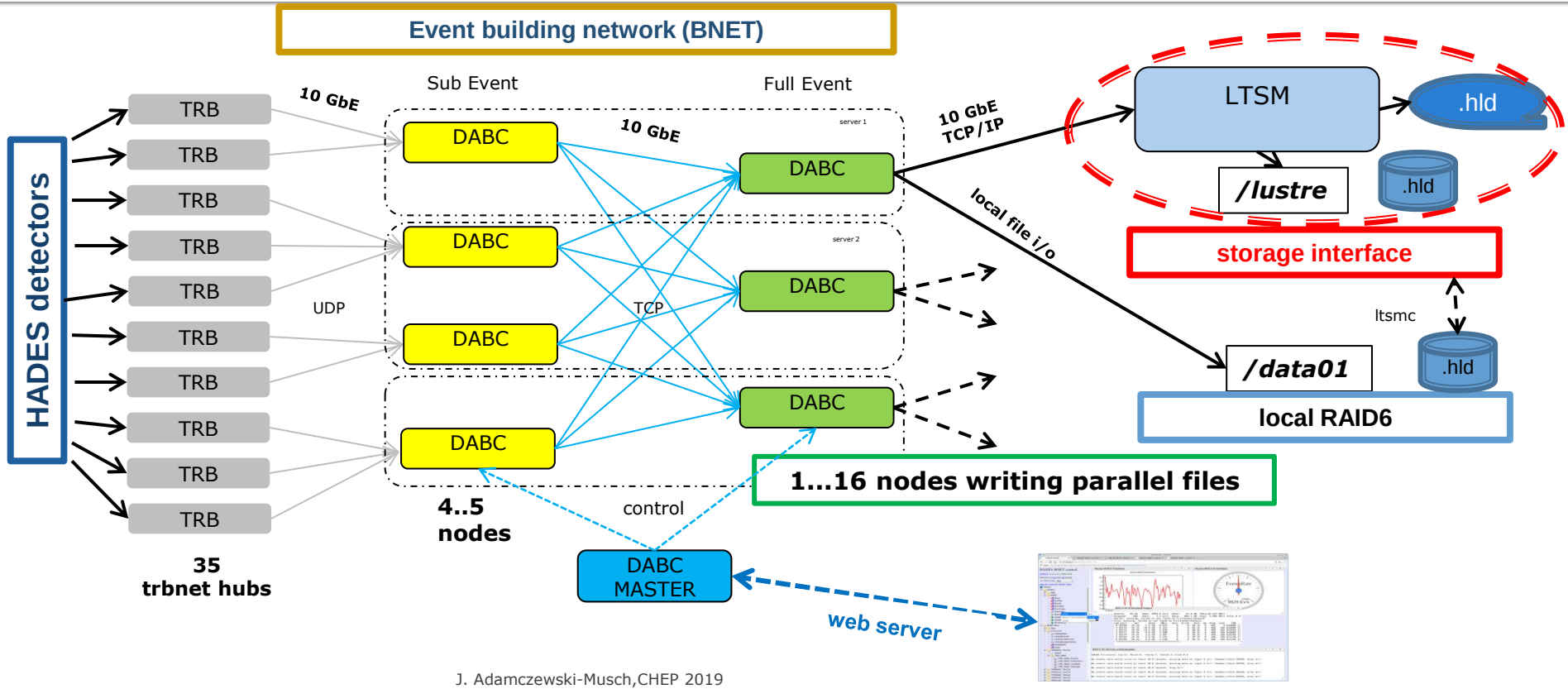
HADES (High Acceptance Di-Electron Spectrometer) experiment at FAIR-0 in March 2019 (Ag-Ag@1.58 AGeV)





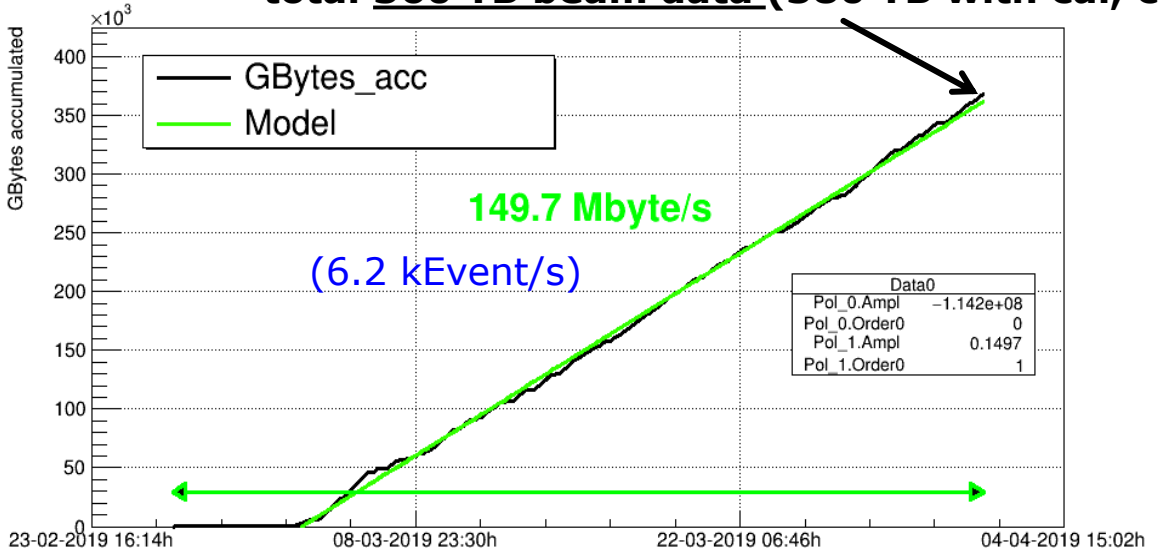
https://www-hades.gsi.de

HADES DAQ March 2019 with LTSM



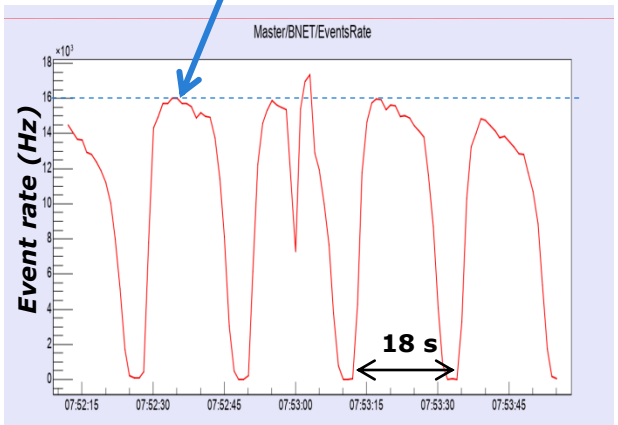
HADES beamtime Ag-Ag@1.58 AGeV: data taking March 2019

total <u>360 TB beam data</u> (380 TB with cal, cosmics etc.)



run netto rates: **208.6 Mbyte/s** (8.7. kEvents/s)
(=> 72% beam time usage)

@spill maximum:
390 Mbyte/s
(16 kEvents/s)



Experiences from MAR19 beamtime (1)

Event builder LTSM socket to TSM server: **mostly OK!** But:

- **TSM server timeout after *1 minute without writing (beam pause during open file)***
  - **Event builder process terminates intentionally**, operator restart required!
  - **started files were lost for archive at first! – about 50 files/day**
  - files could be archived later with **ltsmc script from local eventbuilder disk duplicates**
- reason: **wrong TSM server configuration**! was easily fixed.

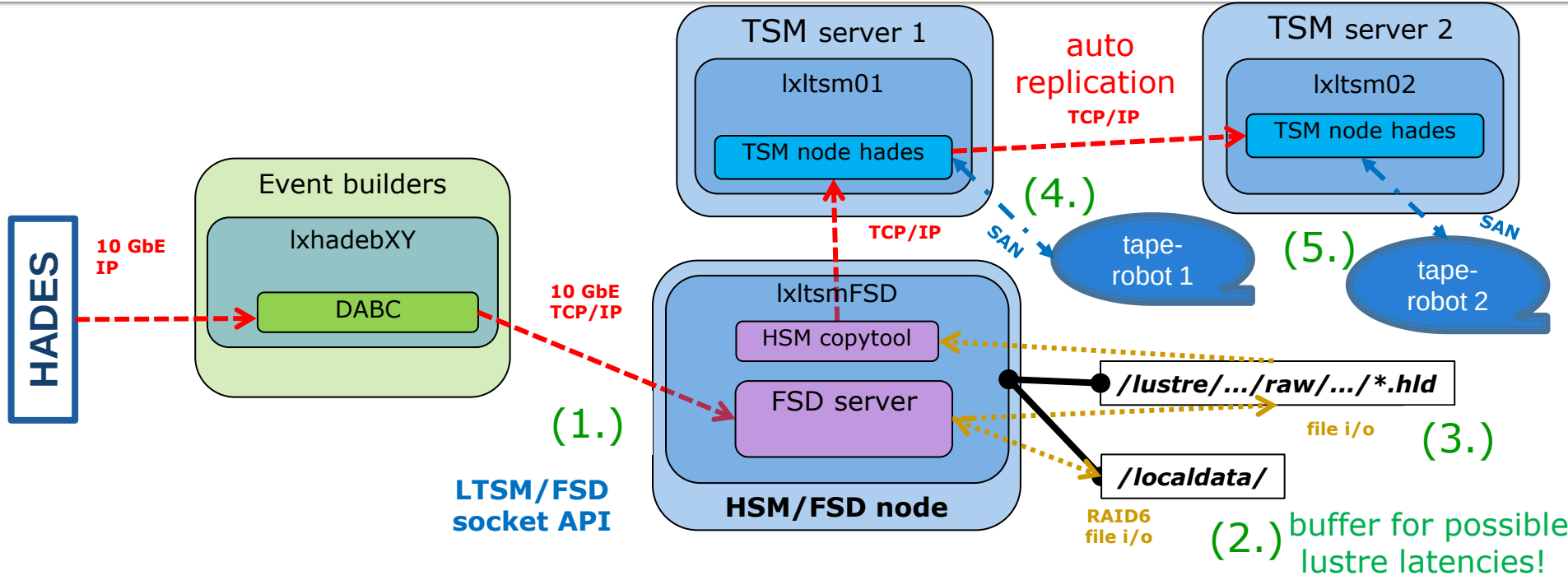
Experiences from MAR19 beamtime (2)

Raw data transfer to /lustre "on time" (within 1 hour ?):

- **provided LTSM methods: *HSM copytool* + requesting script *ltsmsync.sh***
  - ✓ **have proved to work for smaller experiments like TASCA with MBS**
  - **not fast enough for HADES!**
    - ➢ **conflict on TSM server with concurrent tape migration jobs!**
- **HADES workaround: *rsync cronjob* for files from eventbuilder server disks**
  - OK, but sometimes **did affect eventbuilder processes** (cpu io wait!) -> **lost events!**

=> Further LTSM developments required!

Under development: LTSM File System Demon (FSD)



- **files are on lustre first and are archived with copytool subsequently**

Summary and outlook

- LTSM provides interfaces and tools to access IBM TSM storage in connection with the lustre file system
- LTSM has been integrated to GSI DAQ frameworks DABC and MBS
- Experiments HADES, TASCA, R3B,.. have successfully used LTSM for raw data storage during FAIR-0 beam times in 2019
- The LTSM/FSD server is under development for 2020 to provide faster availability of DAQ files on lustre

Thank you!

Bonus slides

LTSM C - API functions

connect/disconnect TSM server

- *int tsm_fconnect(struct login_t *login, struct session_t *session);*
- *void tsm_fdisconnect(struct session_t *session);*

open file

- *int tsm_fopen(const char *fs, const char *fpath, const char *desc, struct session_t *session);*

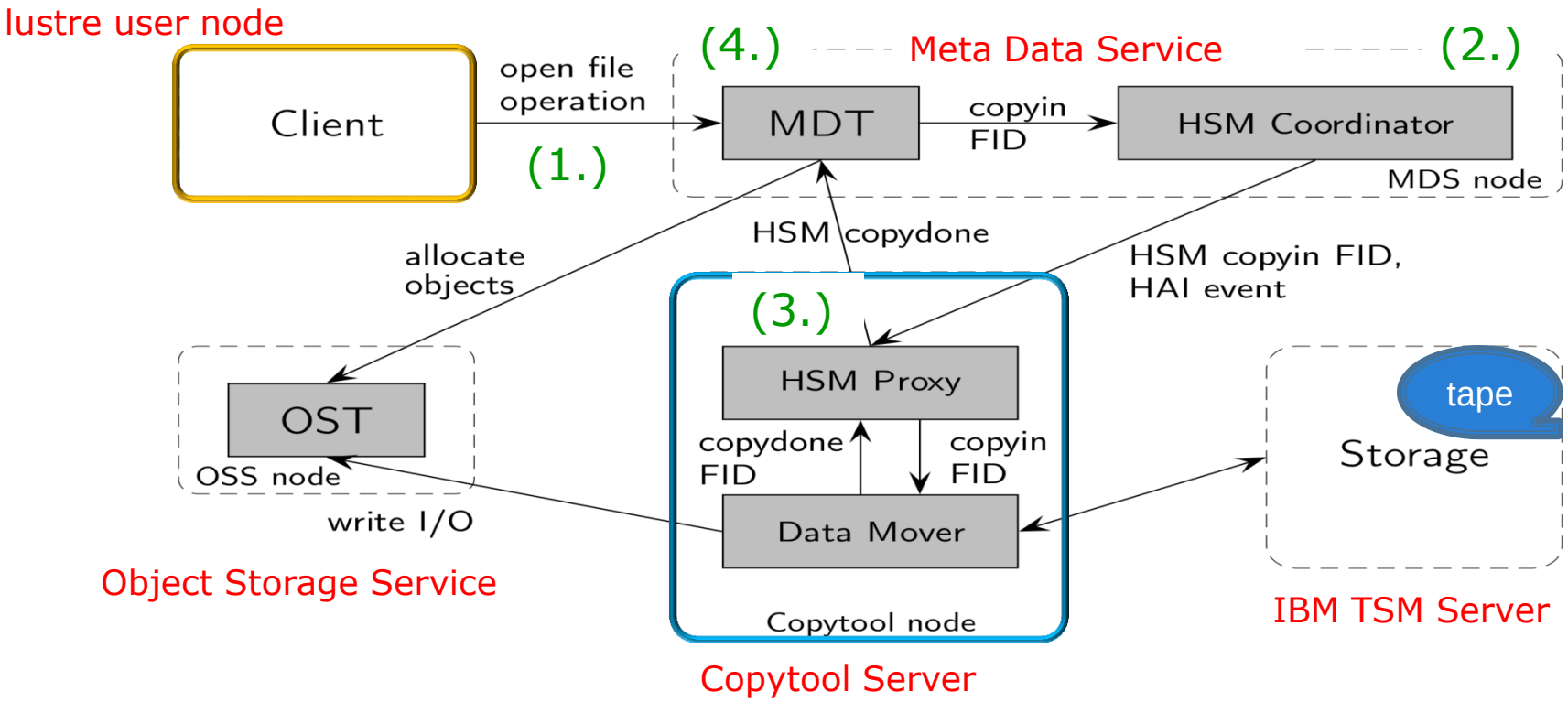
write buffer to file

- *ssize_t tsm_fwrite(const void *ptr, size_t size, size_t nmemb, struct session_t *session);*

close file

- *int tsm_fclose(struct session_t *session);*

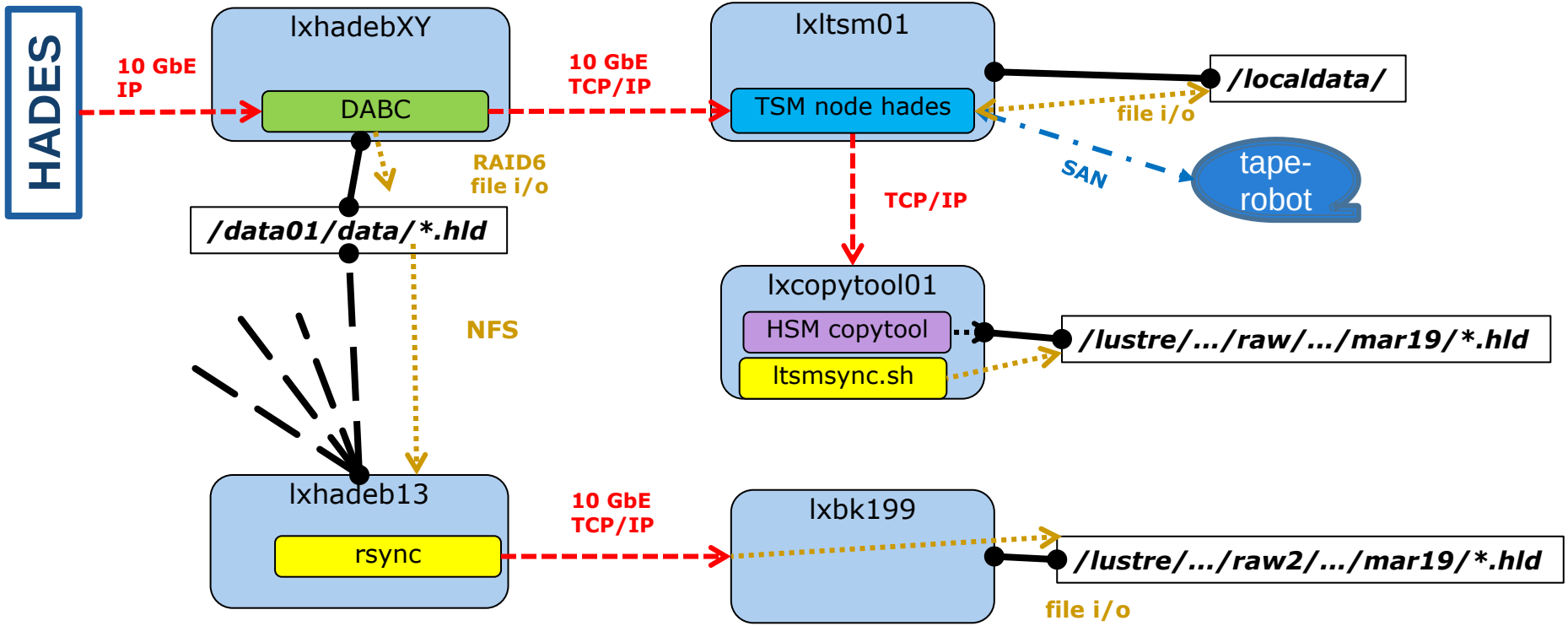
LTSM lustre HSM copy tool



Command line tool ltsmc

- **query** data by *ltsmc –query*
- **retrieve** files by *ltsmc –retrieve*
- manually **archive** local files by *ltsmc –archive*
- allows **higher level scripts** (*examples of HADES*):
  - archive local files with different path on lustre (*archive_ltsm_jul18.sh*)
  - compare local and archived files (*archived_data_ltsm.pl*)
  - archive a set of missing files (*data2tape_ltsm.pl*)

HADES raw data to lustre MAR19: rsync workaround



FSD status and tests

- ✓ **API implemented in DABC**
- ✓ **12 x 80 MB/s socket write test**
  (2 days, random data)
  lxhadeb08,09,10,11 -> lxltsm02 (local RAID)
- transfer to lustre mechanism — TO DO
- HSM sync lustre -> TSM — TO DO
- setup of production server — TO DO
- Test of HADES storage chain
  (EB -> lustre -> TSM) — TO DO

Expected full implementation January 2020

LTSM with FSD API test
(10.-12.09.19 –> 42h , 150TB)